How to Compare Different Range Estimations: A Symmetry-Based Approach

Olga Kosheleva and Vladik Kreinovich

University of Texas at El Paso
El Paso, TX 79968, USA
olgak@utep.edu, vladik@utep.edu

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

1. Outline

- How to compare different range estimators for multivariate functions under uncertainty?
- To answer this question, we analyze which utility functions can be used for this task. Specifically, we:
  - introduce various invariance assumptions,
  - describe the class of all utility functions which satisfy these assumptions, and
  - show how the resulting utility functions can be used to compare different range estimators.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

2. Need for Data Processing

- In many practical situations, we are interested:
  - in the future value of a quantity y
  - or in its current value which is not easy to measure directly.
- In such situations:
  - we find easier-to-measure quantities $x_1, \ldots, x_n$ which are related to y in a known way $y = f(x_1, \ldots, x_n)$,
  - and we use the measured values $\widetilde{x}_i$ of the auxiliary quantities x_i to estimate y as $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.
- This is usually called *data processing*.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

3. Need to Take Measurement Uncertainty Into Account

- Measurements are never absolutely accurate.
- Thus, the measured values $\widetilde{x}_i$ are somewhat different from the actual (unknown) values x_i.
- Therefore, the estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is, in general, different from the desired value $y = f(x_1, \ldots, x_n)$.
- Often, the only information that we have about $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ is the upper bound Δ_i: $|\Delta x_i| \le \Delta_i$.
- In this case, the only information that we know about each x_i is that $x_i \in \mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.
- We therefore need to find the range $\mathbf{y}$ of $f(x_1, \ldots, x_n)$ when $x_i \in \mathbf{x}_i$; this is called *interval computations.*
- The exact computation of this range is known to be NP-hard, so we find an *enclosure* $\mathbf{Y} \supseteq \mathbf{y}$.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

4. Dynamic Case

- Often, we are interested in the values $y(t)$ corresponding to different t (time, location, etc.).
- Different interval techniques lead to different enclosures $\mathbf{Y}(t)$, $\mathbf{Y}'(t)$, ... for $\mathbf{y}(t)$.
- How can we compare these results?
- Of course, if a technique leads to narrower bounds for all t, it is better.
- However, often, one technique is better for some t while another technique is better for other t.
- In this talk, we propose a symmetry-based approach to comparing two methods.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

5. How Preferences Are Described in Decision Theory: The Notion of Utility

- To describe preferences in numerical terms, we select a very bad alternative A_0 and a very good one A_1.
- Then, for each $p \in [0, 1]$, we form a lottery $L(p)$ in which we get A_1 with probability p else A_0.
- When $p \approx 0$, we have $L(p) < A$; when $p \approx 1$, we have $A < L(p)$.
- There exists a threshold value

$$u(A) = \sup\{p : L(p) < A\} \text{ for which:}$$

  - for all $p < u(A)$, we have $L(p) < A$, and
  - for all $p > u(A)$, we have $L(p) > A$.
- This threshold value is called the *utility* of A.
- Here, A is better than A' if and only if $u(A) > u(A')$.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

6. Application to Our Problem

- The numerical value of the utility depends on the selection of the alternatives A_0 and A_1.
- One can show that:
  - if we select a different pair of alternatives $A'_0 < A'_1$,
  - then the new utility values $u'(A)$ is linearly related to $u(A)$: $u'(A) = a \cdot u(A) + b$.
- Let us apply the notion of utility to our problem.
- Usually, we only describe the bounds $\mathbf{Y}(t) = [\underline{Y}(t), \overline{Y}(t)]$ for the values t from a grid: $t_i = t_0 + i \cdot \Delta t$.
- In this case, the utility describing the quality of each estimation has the form $u(\mathbf{Y}(t_0), \mathbf{Y}(t_1), \ldots)$.
- Let us describe common sense properties of preference in terms of utility functions.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

7. Localness Property

- Let us assume that:
  - the two enclosures $\mathbf{Y}(t)$ and $\mathbf{Y}'(t)$ differ only for moments $t \in [\underline{t}, \overline{t}]$, and
  - that $\mathbf{Y}$ is preferred to $\mathbf{Y}'$ ($\mathbf{Y} \geq \mathbf{Y}'$).
- This means that, for the user, on the interval $[\underline{t}, \overline{t}]$, the bounds $\mathbf{Y}(t)$ are better than the bounds $\mathbf{Y}'(t)$.
- This preference should not depend on whatever common bounds $\mathbf{Y}(t) = \mathbf{Y}'(t)$ we have for $t \notin [\underline{t}, \overline{t}]$.
- Formally:
  - if we have bounds $\mathbf{Z}(t)$ and $\mathbf{Z}'(t)$ for which:
    - $\mathbf{Z}(t) = \mathbf{Z}'(t)$ for all $t \notin [\underline{t}, \overline{t}]$, and
    - $\mathbf{Z}(t) = \mathbf{Y}(t)$ and $\mathbf{Z}'(t) = \mathbf{Y}'(t)$ for all $t \in [\underline{t}, \overline{t}]$,
  - then $\mathbf{Z}$ should be preferable to $\mathbf{Z}'$: $\mathbf{Z} \geq \mathbf{Z}'$.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

8. Localness in Terms of Utility

- It is known that preferences are local if and only if $u(x_1, \ldots, x_n)$ is:
  - *additive* $u(x_1, \ldots, x_n) = u_1(x_1) + \ldots + u_n(x_n)$ for some $u_i(x_i)$; or
  - *multiplicative* $u(x_1, \ldots, x_n) = u_1(x_1) \cdot \ldots \cdot u_n(x_n)$.
- In our case, we maximize $\sum_i u(\mathbf{Y}(t_i), t_i)$ or $\prod_i u(\mathbf{Y}(t_i), t_i)$, where $u_i(\mathbf{Y}(t_i)) \stackrel{\text{def}}{=} u(\mathbf{Y}(t_i), t_i)$.
- Maximizing the product is equivalent to maximizing its logarithm – which is a sum.
- So, we always maximize the sum.
- When $\Delta t \to 0$, the sum tends to an integral, so we maximize the integral $\int u(\mathbf{Y}(t), t)\, dt$.
- So, to specify preferences, we must find an appropriate function $u(\mathbf{Y}, t)$.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

9. Other Common-Sense Properties

- Small changes in time t and enclosure $\mathbf{Y}(t)$ should lead to small changes in utility.
- It is therefore reasonable to require that the function $u(\mathbf{Y}, t)$ is smooth.
- Specifically, we require that this function is at least twice differentiable.
- Preferences should not change:
  - if we simply change the unit for measuring time $t \to \lambda \cdot t$ (e.g., from minutes to seconds),
  - or change the starting point $t \to t + t_0$.
- Similarly, we can use different starting points and different units for describing the quantity y.
- Thus, the preference relation should not change if we replace y with $y + y_0$ or with $\lambda \cdot y$.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

10. Definitions: Localness and t-Invariance

- By an *additive utility function*, we mean an expression of the type $u(\mathbf{Y}) = \int u(\mathbf{Y}(t), t)\, dt$, where $u \in C^2$.
- By a *multiplicative utility function*, we mean an expression of the type $u(\mathbf{Y}) = \exp(\int u(\mathbf{Y}(t), t)\, dt)$, $u \in C^2$.
- By a *utility function*, we mean either an additive utility function or a multiplicative utility function.
- $u(\mathbf{Y})$ and $u'(\mathbf{Y})$ are *equivalent* if there exist $a > 0$ and b for which $u'(\mathbf{Y}) = a \cdot u(\mathbf{Y}) + b$ for all $\mathbf{Y}(t)$.
- By a *t-rescaling*, we mean a transformation $T_{\lambda,t_0}(t) = \lambda \cdot t + t_0$ for some $\lambda > 0$ and t_0.
- For each function $\mathbf{Y}(t)$, by $\mathbf{Z} = T_{\lambda,t_0}(\mathbf{Y})$, we mean $\mathbf{Z}(t) \stackrel{\text{def}}{=} \mathbf{Y}(\lambda \cdot t + t_0)$.
- We say that $u(\mathbf{Y})$ is *t-invariant* if for every $\lambda > 0$ and t_0, $u(T_{\lambda,t_0}(\mathbf{Y}))$ is equivalent to $u(\mathbf{Y})$.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

11. Definitions: y-Invariance

- By a *y-rescaling*, we mean a transformation $Y_{\lambda,y_0}(y) = \lambda \cdot y + y_0$ for some $\lambda \neq 0$ and y_0.
- *Comment.*
  - for time, the direction is fixed;
  - in contrast, for many quantities y, there is no fixed direction;
  - e.g.: electric charge q can be $q > 0$ or $q < 0$; formulas do not change if we switch the signs;
  - thus, for y-invariance, we require $\lambda \neq 0$ and not $\lambda > 0$.
- For each function $\mathbf{Y}(t) = [\underline{Y}(t), \overline{Y}(t)]$, by $\mathbf{Z} = Y_{\lambda,y_0}(\mathbf{Y})$, we mean $\mathbf{Z}(t) \stackrel{\text{def}}{=} [\lambda \cdot \underline{Y}(t) + y_0, \lambda \cdot \overline{Y}(t) + y_0]$.
- We say that $u(\mathbf{Y})$ is *y-invariant* if for every $\lambda \neq 0$ and y_0, $u(Y_{\lambda,y_0}(\mathbf{Y}))$ is equivalent to $u(\mathbf{Y})$.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

12. Main Result

- **Proposition:** *Every t-invariant and y-invariant utility function is equivalent:*
  - *either to a functional* $\int \left(\overline{Y}(t) - \underline{Y}(t)\right)^p \, dt$ *for some real number* $p > 0$,
  - *or to the functional* $\int \ln\left(\overline{Y}(t) - \underline{Y}(t)\right) \, dt$.
- *Discussion:*
  - For $p = 1$, we select a method with the smallest average width.
  - For $p = 2$, we select a method based on the mean square width.
  - For $p \to \infty$, we select a method with the smallest worst-case width $\max\limits_t \left(\overline{Y}(t) - \underline{Y}(t)\right)$.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

13. How to Compare Estimates Requiring Different Computation Time T?

- In this case, we need to explicitly describe the dependence of the utility value on T.
- T can also be described by using different units of time.
- By a T-*rescaling*, we mean a transformation $T_{\lambda,y_0}(T) = \lambda \cdot T + T_0$ for some $\lambda > 0$ and T_0.
- We say that $u(\mathbf{Y}, T)$ is T-*invariant* if for every $\lambda > 0$ and T_0, $u(\mathbf{Y}, \lambda \cdot T + T_0)$ is equivalent to $u(\mathbf{Y}, T)$.
- **Proposition.** *Every t-invariant, y-invariant, and T-invariant utility function is equivalent:*
  - *either to a functional $T^{-q} \cdot \int \left(\overline{Y}(t) - \underline{Y}(t)\right)^p \, dt$ for some real number $p > 0$ and $q > 0$,*
  - *or to $T^{-q} \cdot \int \ln\left(\overline{Y}(t) - \underline{Y}(t)\right) \, dt$ for some $q > 0$.*

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

14. Auxiliary Result: How to Compare Different Estimations for the Pareto-Optimal Front

- In many practical problems, we have several objective functions $f_1(x), \ldots, f_n(x)$.
- In this situation, it makes sense to dismiss a solution x if there exists a better (dominating) solution x':

$$f_i(x') \le f_i(x) \text{ for all } i \text{ and } f_i(x') < f_i(x) \text{ for some } i.$$

- The set of all non-dominated alternatives is known as *Pareto optimal front*.
- For $n = 2$, this front can be described by a function $y_2 = f(y_1)$.
- There exist several different techniques for estimating the front. Which one should we select?
- We show that for $n = 2$, a symmetry-based approach can help us in selecting the best estimation method.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

15. Comparing Pareto Front Estimators

- By an *additive utility function*, we mean an expression of the type $u(f) = \int u(f(y_1), y_1)\, dy_1$, where $u \in C^2$.
- By a *multiplicative utility function*, we mean an expression of the type $u(f) = \exp(\int u(f(y_1), y_1)\, dy_1), u \in C^2$.
- By a *utility function*, we mean either an additive utility function or a multiplicative utility function.
- We say that $u(f)$ and $u'(f)$ are *equivalent* if there exist $a > 0$ and b for which $u'(f) = a \cdot u(f) + b$ for all $f(y_1)$.
- By a y_1-*rescaling*, we mean a transformation $F_{\lambda, y_0}(y_1) = \lambda \cdot y_1 + y_0$ for some $\lambda > 0$ and y_0.
- For each $f(y_1)$, by $g = F_{\lambda, y_0}(f)$, we mean

$$g(y_1) \stackrel{\text{def}}{=} f(\lambda \cdot y_1 + y_0).$$

- We say that $u(f)$ is y_1-*invariant* if for every $\lambda > 0$ and y_0, the functional $u(F_{\lambda, y_0}(f))$ is equivalent to $u(f)$.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

16. Comparing Pareto Front Estimators (cont-d)

- By a y_2-*rescaling*, we mean a transformation $S_{\lambda,y_0}(y_2) = \lambda \cdot y_2 + y_0$ for some $\lambda > 0$ and y_0.
- For each $f(y_1)$, by $g = D_{\lambda,y_0}(f)$, we mean

$$h(y_1) \stackrel{\text{def}}{=} \lambda \cdot f(y_1) + y_0.$$

- We say that $u(f)$ is y_2-*invariant* if for every $\lambda > 0$ and y_0, $u(S_{\lambda,y_0}(f))$ is equivalent to $u(f)$.
- **Proposition.** *Every y_1-invariant and y_2-invariant utility function is equivalent to the functional $\int f(y_1)\,dy_1$.*
- So, we should select a method for which the area under the curve $y_2 = g(y_1)$ is the smallest.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

17. Conclusion

- How can we compare range estimators for multivariate functions $y = f(x_1, \ldots, x_n, t)$ under uncertainty?
- According to decision theory, an alternative is better if its utility is larger.
- Thus, to compare different range estimators, we need to find appropriate utility functions.
- It is reasonable to require that the corresponding ordering does not change:
  - if we select different starting points and different units
  - for measuring y and for measuring time t.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

18. Conclusion (cont-d)

- It is also reasonable to require the *localness* property:
  - if replacing an estimate $\mathbf{Y}(t)$ with $\mathbf{Y}'(t)$ on $[\underline{t}, \overline{t}]$ improves the estimation,
  - then a similar replacement should improve no matter what are the values $\mathbf{Y}(t)$ outside this interval.
- Then, the comparison reduces to comparing $V \stackrel{\text{def}}{=} \int \ln(w(t))\,dt$ or $V \stackrel{\text{def}}{=} \int (w(t))^p\,dt$ for some $p > 0$.
- Here, $w(t)$ is the width of the interval $\mathbf{Y}(t)$.
- If we take into account computation time T, then we should compare the values $T^{-q} \cdot V$ for some $q > 0$.
- A similar result holds for comparing different estimators for the Pareto-optimal front.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

19. Acknowledgments

This work was supported in part by the National Science Foundation grants:

- HRD-0734825,
- HRD-1242122, and
- DUE-0926721.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

20. Appendix: Proof

- For an additive utility, time-shift invariance means that for every t_0, there exist $a(t_0)$ and $b(t_0)$ for which

$$\int u(\mathbf{Y}(t), t + t_0)\, dt = a(t_0) \cdot \int u(\mathbf{Y}(t), t)\, dt + b(t_0).$$

- Taking a variational derivative of this equality over $\mathbf{Y}(t)$, we conclude that for every t, t_0, and $\mathbf{Y}$, we have

$$D(t + t_0) = a(t_0) \cdot D(t), \text{ where } D(t) \stackrel{\text{def}}{=} \frac{\partial u}{\partial \underline{Y}}(\mathbf{Y}, t).$$

- A similar functional equation holds for a partial derivative relative to $\overline{Y}$.
- Since we assumed that the function $u(\mathbf{Y}, t)$ is twice differentiable, its derivative $D(t)$ is differentiable.
- Thus, $a(t_0) = \dfrac{D(t + t_0)}{D(t)}$ is also differentiable, as a ratio of two differentiable functions.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

21. Proof: Slide 2

- Differentiating the equality $D(t+t_0) = a(t_0)\cdot D(t)$ relative to t_0 and setting $t_0 = 0$, we conclude that

$$\frac{dD}{dt} = a\cdot D, \text{ where } a \stackrel{\text{def}}{=} \frac{da}{dt_0}_{|t_0=0}.$$

- Hence, $D(t) = C\cdot \exp(a\cdot t)$.
- Invariance relative to $t \to \lambda\cdot t$ means that for every $\lambda > 0$, there exist values $A(\lambda)$ and $B(\lambda)$ for which

$$\int u(\mathbf{Y}(t), \lambda\cdot t)\,dt = A(\lambda)\cdot \int u(\mathbf{Y}(t), t)\,dt + B(\lambda).$$

- Taking a variation derivative of this equality over $\mathbf{Y}(t)$, we conclude that for every t, λ, and $\mathbf{Y}$, we have

$$D(\lambda\cdot t) = A(\lambda)\cdot D(t).$$

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

22. Proof: Slide 3

- The derivative $D(t)$ is differentiable.
- Thus, the function $A(\lambda)$ is also differentiable, as a ratio of two differentiable functions.
- Differentiating the equality $D(\lambda \cdot t) = A(\lambda) \cdot D(t)$ relative to λ and setting $\lambda = 1$, we conclude that

$$t \cdot \frac{dD}{dt} = A \cdot D, \text{ where } A \stackrel{\text{def}}{=} \frac{dA}{d\lambda}_{|\lambda=1}.$$

- For $D(t) = C \cdot \exp(a \cdot t)$, this is only possible when $a = 0$.
- Thus, the partial derivative $D(t) = \dfrac{\partial u}{\partial \underline{Y}}(\mathbf{Y}, t)$ does not depend on t at all.
- A similar statement holds for the partial derivative relative to $\overline{Y}(t)$.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

23. Proof: Slide 4

- Thus, the utility function is equivalent to an expression $\int u(\mathbf{Y}(t))\,dt$ in which $u(\mathbf{Y}(t))$ does not explicitly depend on t.
- Instead of a function $u(\underline{Y}, \overline{Y})$, it is convenient to consider an equivalent function $v(m, w)$, where:
  - $m \stackrel{\text{def}}{=} \dfrac{\underline{Y} + \overline{Y}}{2}$ is the midpoint of the interval $\mathbf{Y}$,
  - $w \stackrel{\text{def}}{=} \overline{Y} - \underline{Y}$ is the width of this interval, and

  – $v(m, w) \stackrel{\text{def}}{=} u\left(m - \dfrac{w}{2}, m + \dfrac{w}{2}\right)$.
- The advantage is that under y-shifts $y \to y + y_0$, only the midpoint changes: $m \to m + y_0$.
- Thus, invariance with respect to y-shifts means that

$$\int v(m(t)+y_0, w(t))\,dt = a(y_0)\cdot\int v(m(t), w(t))\,dt + b(y_0).$$

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

24. Proof: Slide 5

- Differentiating this equality relative to $m(t)$, for the corresponding derivative $D(m)$, we have

$$D(m + y_0) = a(y_0) \cdot D(m).$$

- Similarly to the case of time shifts, this implies that $D(m)$ exponentially depends on m.
- If we take into account scale-invariance, this means there is no dependence on $m(t)$.
- Therefore, the utility function takes the form $\int v(w(t))\,dt$, where $w(t) = \overline{Y}(t) - \underline{Y}(t)$.
- In terms of the width $w(t)$, the transformation $y \to \lambda \cdot y$ leads to $w(t) \to \lambda \cdot w(t)$.
- Thus, scale-invariance implies that

$$\int v(\lambda \cdot w(t))\,dt = A(\lambda) \cdot \int v(w(t))\,dt + B(\lambda).$$

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

25. Proof: Slide 6

- For the corresponding variational derivatives $D(w)$, we get $\lambda \cdot D(\lambda \cdot w) = A(\lambda) \cdot D(w)$.
- Hence $D(\lambda \cdot w) = c(\lambda) \cdot D(w)$, where we denoted

$$c(\lambda) \stackrel{\text{def}}{=} \frac{A(\lambda)}{\lambda}.$$

- Here, the function $c(\lambda)$ is differentiable as a ratio of two differentiable functions.
- Differentiating the equality $D(\lambda \cdot w) = c(\lambda) \cdot D(w)$ with respect to λ and taking $\lambda = 1$, we get

$$w \cdot \frac{dD}{dw} = c \cdot D, \text{ where } c \stackrel{\text{def}}{=} \frac{dc}{d\lambda}_{|\lambda=1}.$$

- Moving all the terms related to D to one side and all the terms related to w to another side, we get

$$\frac{dD}{D} = c \cdot \frac{dw}{w}.$$

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .

26. Proof: Last Slide

- Integrating both sides, we get $\ln(D) = c \cdot \ln(w) + c_0$ for some constant c_0.
- Thus, for $D = \exp(\ln(D))$, we get $D(w) = \text{const} \cdot w^c$.
- Let us now recall that $D(w) = \dfrac{\partial v}{\partial w}$.
- Thus, to recover the expression for $v(w)$, we must integrate this derivative $D(w)$ with respect to w.
  - For $c \neq -1$, integration leads to the power dependence $v(w) = w^p$ for $p = c + 1$.
  - For $c = -1$, we get the logarithmic dependence.
- Thus, for additive utility functions, the proposition is proven.
- For multiplicative utility functions, the proof is similar.

Need for Data Processing
Need to Take . . .
Dynamic Case
How Preferences Are . . .
Localness Property
Other Common-Sense . . .
Main Result
How to Compare . . .
Auxiliary Result: How . . .